Exploiting Continuous Motion Clues for Vision-Based Occupancy Prediction

Haoran Xu[1,2], Peixi Peng[2,3*], Xinyi Zhang[1,4], Guang Tan[1*], Yaokun Li[1], Shuaixian Wang[1], Luntong Li[2]

[1]School of Intelligent Systems Engineering, Shenzhen Campus of Sun Yat-sen University
[2]Peng Cheng Laboratory
[3]School of Electronic and Computer Engineering, Shenzhen Graduate School, Peking University
[4]Nio Inc.

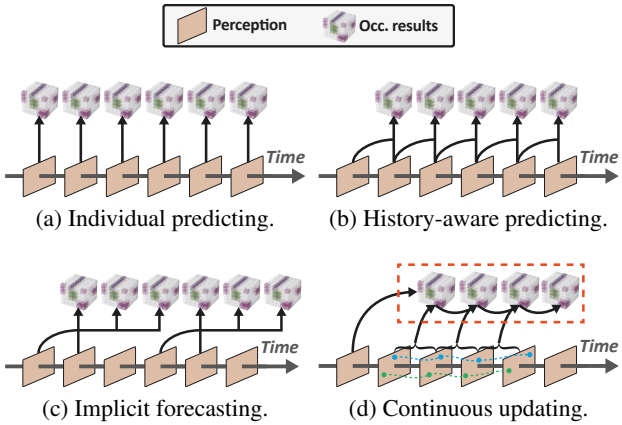
Abstract

Occupancy networks aim to reconstruct the surroundings with occupied semantic voxels. However, frequent object occlusions often occur in dynamic real-world scenarios, which cannot be captured by independent frames. Most existing occupancy networks generate results without explicitly considering past occupancy states and continuous visual changes over time, limiting their temporal accuracy. We tackle it by treating the task from a new continuous updating perspective, which considers historical data and continuous motion clues. We propose a new approach termed Continuous Motion clue exploitation for Occupancy Prediction (CMOP), which incorporates three key designs: (i) Propagator: which forecasts future occupancy states based on historical data; (ii) Tracker: which updates the occupancy on a per-frame basis using dynamic visual motion information; and (iii) Fuser: which aggregates results from the Propagator and Tracker into more robust and accurate occupancy results. Experiments on several benchmarks demonstrate that CMOP outperforms state-of-the-art baselines.

Introduction

3D occupancy networks offer a comprehensive understanding of complex and dynamic environments by reconstructing the surroundings with fine-grained geometry and semantics (Cao and de Charette 2022; Huang et al. 2023; Li et al. 2023b; Pan et al. 2023). This task has found critical applications in various fields, including autonomous vehicles (Wei et al. 2023b; Ma et al. 2024b) and robotic navigation (Liu, Wang, and Yang 2024; Li et al. 2023a). Recently, it has been the focus of extensive research in both academia and industry (Inc. 2020, 2021; Corp. 2024).

To achieve precise and fine-grained quality of scene reconstruction, most off-the-shelf occupancy models (Huang et al. 2023; Wei et al. 2023b; Cao and de Charette 2022; Li et al. 2023b) typically process each frame independently to identify and reconstruct objects within images, as shown in Figure 1a. As illustrated in Figure 2, the first row shows consecutive perceptions from a camera on the ego-vehicle, while the second row displays the corresponding occupancy results. It is clear that a vehicle obstructs the view of a pedestrian at time t_2, resulting in missing occupied voxels and reduced temporal accuracy during this period. These methods follow a "**detection**" scheme, producing occupancy results independently either from individual frames or by using motion clues derived from multiple frames. However, in dynamic scenarios, several objects may be occluded by others or cannot be observed clearly at some time, and this may cause inaccurate occupancy predictions.



(a) Individual predicting. (b) History-aware predicting. (c) Implicit forecasting. (d) Continuous updating.

Figure 1: Illustration of different inference schemes of occupancy networks. (a) Predicting individual occupancy using the current frame; (b) Predicting current occupancy with the help of multiple historical frames; (c) Forecasting multiple future occupancy results through implicit motion extraction; (d) Continuously updating occupancy results using both historical results and explicit motion clues.

Since the dynamic scene is changed continuously, the historical information and temporal motion should be useful cues to handle the above challenge. Several methods (Huang and Huang 2022; Li et al. 2023c) attempted to leverage historical data to enhance the accuracy of current occupancy inference, as shown in Figure 1b. In addition, some recent spacetime (4D) methods (Ma et al. 2024a; Khurana et al. 2023; Mersch et al. 2022) have attempted to implicitly learn motion features from past frames to simultaneously forecast future occupancy states in an end-to-end manner (termed implicit forecasting, see Figure 1c). However, these methods rely on historical observations to predict current (or future) occupancy in a latent manner and fail to explicitly use past

*Corresponding authors.

3D occupancy results and motion features as priors, which could directly enhance 3D occupancy predictions.

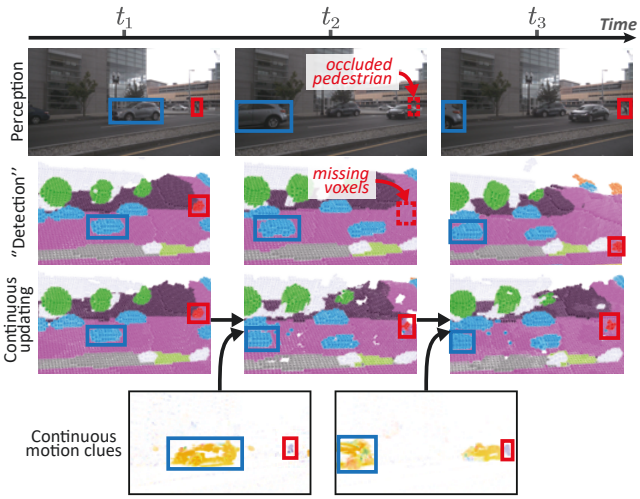


Figure 2: A typical example highlighting the importance of continuous updating. Our method can enhance temporal accuracy using both historical occupancy results and continuous visual changes.

Hence, we introduce a new vision-based Occupancy Prediction approach by exploiting Continuous Motion clues (CMOP) to address the above problems from a new "**continuous updating**" perspective. The key idea is to leverage both historical data and live motion clues to predict up-to-date occupancy state, in a fast and adaptive way. As shown in the last two rows of Figure 2, our CMOP go beyond considers only historical occupancy results and also takes into account continuous motion clues (optical flow (Farnebäck 2003; Beauchemin and Barron 1995)). Optical flow provides explicit pixel-level guidance, capturing the apparent motion changes in the occupancy results through a vector field. By leveraging this combined information, our CMOP exploits the continuous visual changes and advances occupancy states iteratively, leading to more temporally accurate reconstructions of the surroundings. Specifically, the general pipeline of CMOP comprises four components: **(i)** a standard occupancy inference network (OccInfer) that generates an initial occupancy result; **(ii)** a Propagator module that forecasts future occupancy results based on implicit motion clues inherent in historical data; **(iii)** a Tracker module that updates the occupancy on a per-frame basis with explicit dynamic visual motion information; and **(iv)** a Fuser module that adaptively aggregates occupancy results and dynamic motion features at each prediction step.

The main contributions of this paper are three-fold:

- A new occupancy inference scheme that performs in a continuous updating manner.
- A lightweight framework (CMOP) that effectively leverages motion clues from both historical occupancy results and explicit optical flow for more temporally accurate occupancy results.
- Extensive experiments that validate our method could achieve SOTA performances on several benchmarks.

Related Works

Vision-based scene reconstruction

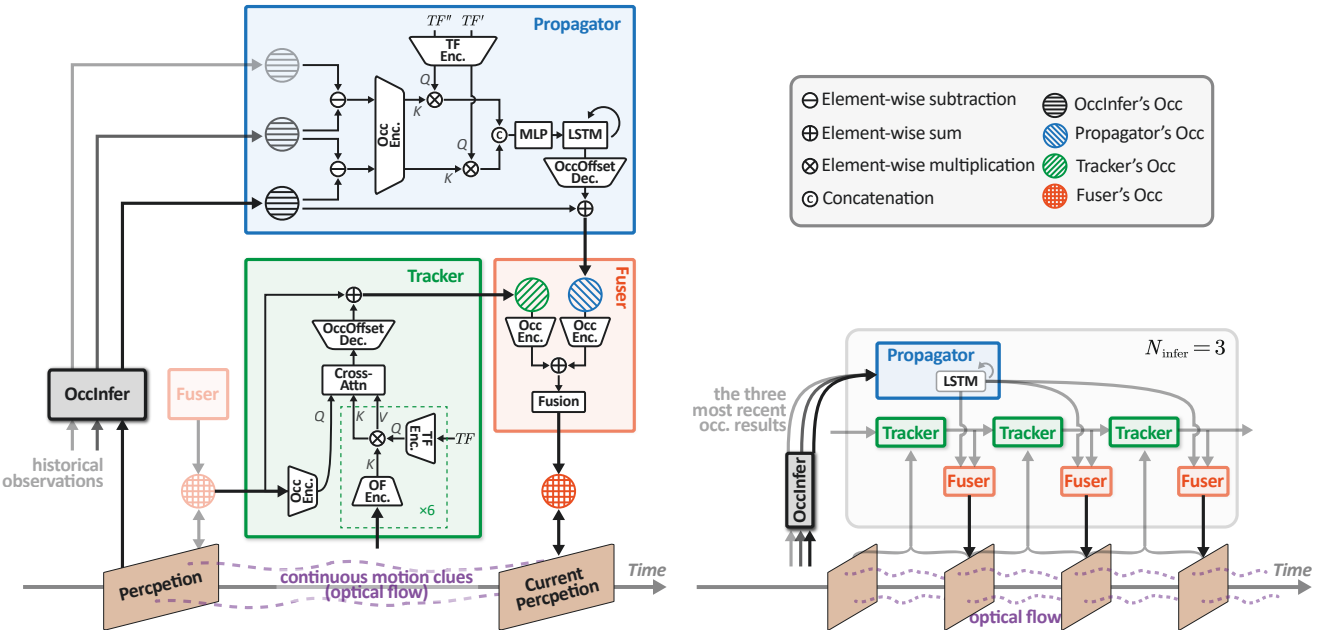
The process of vision-based scene reconstruction primarily involves transforming original multi-camera observations into a more concise and structured format, while retaining the inherent scene geometry. This is accomplished through four possible approaches: **(i)** Depth-based maps (Wei et al. 2023a; Xu et al. 2023; Ju et al. 2023), which explicitly delineate the relative spatial distances between various objects and surfaces; **(ii)** Neural Radiance Fields (NeRF) (Li, Wang, and Tan 2024; Li, Gou, and Tan 2024; Wang et al. 2024), which work as an implicit function to predict the opacity and color field of sampled points in 3D space; **(iii)** Bird's Eye View (BEV) (Li et al. 2022; Yang et al. 2023; Liang et al. 2022), which transforms the perspective into a top-down view, providing insights into the geometric size and spatial location of objects; **(iv)** Volumetric occupancy (Wei et al. 2023b; Huang et al. 2023; Li et al. 2023b), which yields detailed semantic labels associated with each occupied voxel.

3D volumetric scene reconstruction

3D scene reconstruction aims to estimate the geometry, semantics, and spatial structure of objects within a scene, while also define their positions and relationships in 3D space (Cao and de Charette 2022). Both indoor (Sun et al. 2021; Bozic et al. 2021; Gao, Mao, and Liu 2023) and outdoor (Wei et al. 2023a; Huang et al. 2023; Yan et al. 2021) scene reconstruction have achieved great progress, the difference is that the ego-vehicle can move fast and the objects are frequently obscured or appearing. Due to the increasing demand for high-fidelity input visual images and fine-grained outputs, some works (Yu et al. 2023; Tang et al. 2024) aim to optimize the network architecture and the learned latent space. In this paper, we opt to solve the inherent challenge of temporal accuracy through a novel continuous updating perspective.

Occupancy forecasting

Occupancy forecasting aims to predict future occupancy states beyond the current time. The past few years have witnessed the prosperity of occupancy forecasting on the Bird's-Eye-View (BEV) occupancy grids (Casas, Sadat, and Urtasun 2021; Hu et al. 2021; Mahjourian et al. 2022). The advent of spacetime (4D) occupancy forecasting has further advanced the field. Some 4D methods focused on sensor-level prediction (Lu et al. 2021; Mersch et al. 2022), which can be voxelized to future occupancy result, while others directly predict the future rendered occupancy in an end-to-end way (Ma et al. 2024a). However, existing 4D occupancy forecasting methods simultaneously forecast future occupancy states based solely on historical observations, neglecting to explicitly utilize past 3D occupancy results and continuous visual changes as priors. If this combined infor-



(a) Single updating. (b) Expanded updating.

Figure 3: CMOP framework. CMOP can work in the following two ways: (a) Single updating: CMOP acts as a refinement tool with one-step updating ($N_{\text{infer}} = 1$). (b) Expanded updating: CMOP performs continuous updating over $N_{\text{infer}} = 3$ steps. Within CMOP, Propagator takes historical occupancy results generated by OccInfer as input to forecast future occupancy states, Tracker utilizes continuous motion clues (optical flow) to iteratively update occupancy results. Fuser adaptively aggregates results from Propagator and Tracker to generate a more accurate and robust result.

mation is fully utilized, it could directly enhance the temporal accuracy of occupancy predictions.

CMOP Design

Overview

The objective of our method is to perform updating for each time step throughout the inference window of OccInfer based on both historical occupancy results and continuous motion clues. The length of OccInfer's inference window is defined as N_{infer}. Figure 3 illustrates our key designs and two types of running procedures. The left image shows a single updating of CMOP while the right shows an example of extended updating over $N_{\text{infer}} = 3$ steps.

The basic operational workflow of our method is as follows: OccInfer is periodically invoked every N_{infer} steps. When a new inference cycle begins, both Propagator and Tracker also initiate, operating in parallel with OccInfer. Propagator forecasts future occupancy results based on the three most recent historical occupancy results generated by OccInfer. Tracker leverages continuous visual changes to incrementally update occupancy states based on optical flow. Then, Fuser adaptively aggregates results at the time when both Propagator and Tracker are ready to generate a more accurate and robust occupancy.

Notably, as N_{infer} increases, OccInfer can periodically deliver fresh results to minimize accumulated errors. Also, since N_{infer} controls the frequency of calling OccInfer, a larger N_{infer} results in more continuous updating using CMOP, thereby balancing performance and efficiency. For a more generalized symbolic representation, let T and t be the steps for inference and updating processes, respectively. The inference window spans from T_0 to T_{+1}, which includes N_{infer} time steps. Discrete subscripts specify particular steps, with "0" representing the first step in the inference window.

Occupancy propagator

Propagator extracts motion-related features from three consecutive occupancy results generated by OccInfer. Each occupancy result, denoted by $[\tilde{O}^{t_0}_{T_{-3}}, \tilde{O}^{t_0}_{T_{-2}}, \tilde{O}^{t_0}_{T_{-1}}]$, has dimensions of $C \times W \times L \times H$. These dimensions represent the number of semantic categories, width, length, and height, respectively. Initially, residual changes are identified through the element-wise subtraction of adjacent occupancies. Following this, a shared Occ Encoder (Φ^{occ}_{prop}) is executed to learn a lower-dimensional motion representation. The encoder uses four convolution (CNN) layers with ReLU nonlinearity to downsample residual occupancies into abstracted feature map, denoted by $F_o \in \mathbb{R}^{c \times w \times l \times h}$. This map is then further processed by a fully-connected (FC) layer followed by layer nomalization (LN) and ReLU nonlinearity to reduce the dimension of F_o into $f_o \in \mathbb{R}^{N_{\text{mot}} \times 1}$.

In practice, given the known relative translation and rotation matrices between two time steps, we compose these matrices along channel dimensions to form a new transformation matrix, denoted by $TF \in \mathbb{R}^{N_{tf} \times 1}$. This matrix

indicates the ego-vehicle's 6-Degrees of Freedom (6-DOF) changes over time, which can be treated as the query to distill the motion feature f_o. Therefore, a TF Encoder (Φ^{tf}_{prop}) is used to project TF into an attention score $fs \in \mathbb{R}^{N_{\text{mot}} \times 1}$ with Sigmoid activation. Then, the historical motion feature can be obtained as:

$$M_{prop} = \Phi^{mlp}_{prop} \Big(\Big[\Phi^{occ}_{prop}[\tilde{O}^{t_0}_{T_{-2}} - \tilde{O}^{t_0}_{T_{-3}}] \otimes \Phi^{tf}_{prop}[TF''] \Big] \copyright \Big[\Phi^{occ}_{prop}[\tilde{O}^{t_0}_{T_{-1}} - \tilde{O}^{t_0}_{T_{-2}}] \otimes \Phi^{tf}_{prop}[TF'] \Big] \Big), \tag{1}$$

where Φ^{mlp}_{prop} is used to aggregate motion features from two intervals. $\otimes$ and $\copyright$ are element-wise multiplication and concatenate operations, respectively. TF'' and TF' represent the transformation matrices linking T_{-3} to T_{-2} and T_{-2} to T_{-1}, respectively.

Based on the extracted historical motion features M_{prop}, Propagator uses the Long Short-Term Memory (LSTM) network (Van Houdt, Mosquera, and Nápoles 2020) to predict subsequent motion features $M^{t_i}_{prop}$:

$$h^{t_{i+1}} = \text{LSTM}\left[M^{t_i}_{prop}; M_{prop}; h^{t_i}\right], \tag{2}$$

$$M^{t_{i+1}}_{prop} = \text{MLP}_{prop}\left[h^{t_{i+1}}\right], \tag{3}$$

where the first input $M^{t_0}_{prop}$ for the LSTM is set to M_{prop}. Then, $M^{t_i}_{prop}$ is updated in sequence with an continuous time step t_i using Eqs. (2) and (3).

The Occ Offset Decoder (Υ_{prop}) with a symmetric structure of the Occ Encoder is utilized to lift the future motion features to 3D occupancy offsets $\Delta OP^{t_{i+1}}_T$:

$$\Delta OP^{t_{i+1}}_T = \Upsilon_{prop}\left[M^{t_{i+1}}_{prop}\right]. \tag{4}$$

These offsets are progressively overlaid on the newly obtained occupancy $\tilde{O}^{t_0}_{T_{-1}}$ to generate the final propagation results, denoted by $OP^{t\in\{t_0,\cdots,t_5\}}_T$:

$$OP^{t_i}_T = \tilde{O}^{t_0}_{T_{-1}} + \sum_i \Delta OP^{t_{i+1}}_T. \tag{5}$$

Occupancy tracker

The Tracker module is responsible for handling incremental occupancy updating, using the optical flow (OF). As shown in Figure 3, CMOP takes as input the latest updating occupancy result at time step t_i and its subsequent optical flow $OF^{t_i:t_{i+1}}$. To be specific, We employ the Occ Encoder Φ^{occ}_{track}, which has the same structure to that in Propagator, to abstract occupancy into $fo^{t_i}_{track}$. Then, during the process from time step t_i to t_{i+1}, we perform downsampling on the vehicle's multi-camera images under the coefficient $0 < \mathcal{F} < 1$, followed by per-pixel motion estimation (Bradski and Kaehler 2008), thereby acquiring the optical flow $OF^{t_i:t_{i+1}}$. Subsequently, we use the OF Encoder (Φ^{of}_{track}) to extract OF into an transient motion guidance feature $fd \in \mathbb{R}^{N_{mot} \times 1}$. The OF Encoder includes three CNN layers and ReLU nonlinearity. The feature fd then serves as key in the TF distillation operation:

$$\overline{M}^{t_i}_{track} = \Phi^{of}_{track}[OF^{t_i:t_{i+1}}] \otimes \Phi^{tf}_{track}[TF^{t_i}_{T_0}], \tag{6}$$

where Φ^{tf}_{track} has the same structures as that in Propagator. We use independent encoders for each camera's input to enhance performance, thus producing $M^{t_i}_{track} \in \mathbb{R}^{N_{mot} \times 6}$.

Next, we leverage Cross-Attention to extract the interactive features of the current Occupancy across six cameras. Specifically, $\overline{M}^{t_i}_{track}$ acts as the key and value, while $fo^{t_i}_{track}$ serves as the query. The output of Cross-Attention is denoted by $M^{t_i}_{track}$. In this way, the implicit correlation between occupied voxels and motion features can be learned. Then, the same decoding structure utilized in Propagator is deployed to lift the motion feature $M^{t_i}_{track}$ into the 3D occupancy offsets $\Delta OT^{t_{i+1}}_{T_0}$:

$$\Delta OT^{t_{i+1}}_{T_0} = \Upsilon_{track}\left[M^{t_i}_{track}\right]. \tag{7}$$

The occupancy offsets are aggregated into the current occupancy $OT^{t_i}_{T_0}$ to generate the updated result:

$$OT^{t_{i+1}}_{T_0} = OT^{t_i}_{T_0} + \Delta OT^{t_{i+1}}_{T_0}. \tag{8}$$

Note that, due to error accumulation, the initial occupancy input for Tracker in a new inference window is a fused occupancy derived from Fuser (elaborated next).

Occupancy fuser and optimization

Propagator relies on historical data to forecast future occupancy, while Tracker performs step-wise updating guided by transient visual changes. Therefore, these two results can be further merged to achieve a more robust and accurate result. We use lightweight CNNs to adaptively learn the fusion weights of their respective results at every time step:

$$\hat{O}^{t_i}_{T_0} = \text{W}_{final}\left[\text{W}_{prop}[OP^{t_i}_{T_0}] \oplus \text{W}_{track}[OT^{t_i}_{T_0}]\right], \tag{9}$$

where W_{prop}, W_{track}, and W_{final} are learnable CNN parameters of fusion weights. As depicted in Figure 3, for the first-step fusion of Propagator and Tracker, the result from Tracker is specified as the occupancy from the final time step in the preceding window.

By following the convention (Wei et al. 2023b; Cao and de Charette 2022; Yu et al. 2020), cross-entropy loss ($\mathcal{L}_{\text{ce}}$), and semantic and geometry affinity losses ($\mathcal{L}_{\text{sem}}$ and $\mathcal{L}_{\text{geo}}$) are employed to supervise the learning of forecasted occupancy, denoted by $\mathcal{L} = \mathcal{L}_{\text{ce}} + \mathcal{L}_{sem} + \mathcal{L}_{\text{geo}}$. Then, Propagator, Tracker, and Fuser modules can be jointly trained in an end-to-end manner by optimizing occupancy results at every updating time step within the current inference window:

$$\mathcal{L}_{\text{total}} = \frac{1}{N_{\text{infer}}} \sum_i \lambda_{sep}(\mathcal{L}(OP^{t_i}_{T_0}, \overset{*}{O}{}^{t_i}_T) + \mathcal{L}(OT^{t_i}_{T_0}, \overset{*}{O}{}^{t_i}_T)) + \lambda_{fuse}\mathcal{L}(\hat{O}^{t_i}_{T_0}, \overset{*}{O}{}^{t_i}_T), \tag{10}$$

where λ_{sep} and λ_{fuse} denote the coefficients associated with the loss from the separate and fused results, respectively.

Experiments

Experimental settings

Targets and evaluation metrics: The experiments are performed to accomplish the following three targets:

Models	IoU ↑	mIoU ↑	Time (s) ↓
MonoScene	23.96	7.31	0.87
Atlas	28.66	15.00	0.48
TPVFormer	30.86	17.10	0.32
BEVFormer	30.50	16.75	0.34
RenderOcc	29.20	19.00	0.67
SurroundOcc	31.49	20.30	0.34
IMVT3D	31.85	18.88	0.51
Cam4DOcc	33.72	21.73	0.91
BEVDet4D	34.92	23.91	0.72
MonoScene	38.42	16.38	0.90
+CMOP	**+14.46**	**+9.07**	+3.4%
SurroundOcc	40.94	22.71	0.37
+CMOP	**+9.45**	**+2.41**	+8.8%
BEVDet4D	42.83	24.01	0.75
+CMOP	**+7.91**	**+0.10**	+4.1%

Table 1: Comparison of occupancy prediction processed in a frame-by-frame fashion using different SOTA methods. All methods are trained with the same GT for a fair comparison. Enhanced results are **bolded** and reduced results are underlined. The best performance of each metric is highlighted with a gray background.

- To validate the occupancy accuracy of CMOP when compared to the state-of-the-art (SOTA) methods, the widely-used metrics (Cao and de Charette 2022; Wei et al. 2023b), i.e., intersection over union (IoU) and mean IoU (mIoU) of all semantic classes, are used for evaluating the accuracy of all occupied voxels. The results are provided in Table 1, where $N_{\text{infer}} = 1$.
- To assess the temporal accuracy of different inference schemes, we introduce the averaged IoU ($\overline{\text{IoU}}$) and averaged F1 score (Goutte and Gaussier 2005) ($\overline{\text{F1}}$) to measure the temporal accuracy of occupancy results across long-duration scenes. These metrics span from the first frame of OccInfer to the end of each scene's horizon. The inference window N_{infer} is set as 3 and 6 frames. The results are provided in Table 2.
- To verify the effectiveness of our key designs in CMOP, we perform ablation studies, as shown in Tables 3 and 4, using $\overline{\text{IoU}}$ and $\overline{\text{F1}}$ metrics when $N_{\text{infer}} = 6$.

A higher value of the metrics indicates a stronger agreement between the predicted occupancy and the GT.

Dataset: The nuScenes benchmark (Caesar et al. 2020) along with dense occupancy ground truth (GT) labels (Wei et al. 2023b) are used for evaluation. There are six camera views, each offering a perception shape of $1600 \times 900 \times 3$. The unlabeled intermediate frames from the "sweeps" folder in the nuScenes dataset are utilized to calculate the optical flow (Bradski 2000). TF matrix, with $N_{tf} = 6$, can be obtained by subtracting the translation and rotation matrices of adjacent frames and then concatenating the results through the channel. The occupancy output shape is fixed at $W = 200, L = 200, H = 16$, with every occupied voxel measuring 0.5 meters. Accordingly, the occupancy scope for both sides width and length ranges from -50 m to 50 m, and the vertical height varies from -5 m to 3 m. The occupied voxels' value signifies the semantic class identifier, which encompasses $C = 17$ distinct classes (class 0 denotes unoccupied voxels). Following the official dataset split rule (Wei et al. 2023b), 700 and 150 scenes are employed for the training and testing phases, respectively. Each scene has a maximum scenario length of 40 frames.

Implementation details: All experiments are conducted on 4 Nvidia A100s, and deployed on the Nvidia Jetson AGX Orin. The downsampled factor $\mathcal{F}$ of optical flow is set to 3 and the dimensions for the latent motion features N_{mot} is set to 100. The abstracted occupancy feature map is fixed at $32 \times 9 \times 9 \times 1$. The tradeoff coefficients of losses $\lambda_{sep,fuse}$ are initially established at 0.4, and 0.2, respectively. To ensure efficient optimization, λ_{sep} is reduced progressively with a step-wise decay rate of 0.99998. Conversely, λ_{fuse} increases, maintaining the sum of all coefficients at 1. More detailed settings can be found in the supplementary material and the publicly available code repository[1].

Comparison with the state-of-the-art

Comparison on occupancy accuracy For this task, we conduct a comparative analysis of CMOP with a variety of SOTA methods, including individual predicting methods, MonoScene (Cao and de Charette 2022), Atlas (Murez et al. 2020), TPVFormer (Huang et al. 2023), BEVFormer (Li et al. 2022), RenderOcc (Pan et al. 2023), SurroundOcc (Wei et al. 2023b), IMVT3D (Ming et al. 2024), and a history-aware method, BEVDet4D (Huang and Huang 2022), and an implicit forecasting method, Cam4DOcc (Ma et al. 2024a). For BEVFormer and BEVDet4D, a 3D segmentation head is appended to the original output to predict semantic occupancy. Besides, since the official Cam4DOcc model forecasts four future occupancy results, we use the first forecasted result for comparison. We selected two "detection" scheme methods for integration with our CMOP: MonoScene, a pioneering baseline, and SurroundOcc, a SOTA method. Additionally, we included BEVDet4D, a history-aware SOTA approach, to ensure a comprehensive evaluation. All methods are trained using the same GT. The results are shown in Table 1.

It is clear that the history-aware BEVDet4D method outperforms all baselines in terms of accuracy but is more time-consuming compared to most individual predicting methods (e.g., SurroundOcc). This is because BEVDet4D enhances the current occupancy inference by incorporating data from both the current and historical frames. On the other hand, Cam4DOcc predicts future frames solely based on historical data, leading to suboptimal performance and increased inference time. Though CMOP introduces a slight increase in inference time of 0.03 seconds due to the additional computations, integrating CMOP into the baseline methods significantly enhances IoU and mIoU by refining their preliminary occupancy results.

[1]https://github.com/kyoran/CMOP

Models	$N_{\text{infer}} = 3$		$N_{\text{infer}} = 6$	
	$\overline{\text{IoU}} \uparrow$	$\overline{\text{F1}} \uparrow$	$\overline{\text{IoU}} \uparrow$	$\overline{\text{F1}} \uparrow$
MonoScene	17.89	29.27	13.34	24.91
SurroundOcc	23.82	38.11	21.10	34.46
BEVDet4D	32.54	44.71	29.33	42.98
Cam4DOcc	30.01	42.53	28.48	40.14
MonoScene	26.16	38.40	22.75	35.94
+CMOP	**+8.27**	**+9.13**	**+9.41**	**+11.03**
SurroundOcc	34.11	49.13	31.56	46.71
+CMOP	**+10.29**	**+11.02**	**+10.46**	**+12.25**

Table 2: Comparison of occupancy prediction across varying inference lengths using different SOTA methods. Enhanced results are **bolded**. The best and second-best performances of each metric are highlighted in gray background.

Comparison on temporal accuracy In this task, we highlight the effectiveness of the continuous updating idea used in CMOP by comparing it with two 3D occupancy prediction methods, MonoScene (Cao and de Charette 2022) and SurroundOcc (Wei et al. 2023b), a history-aware method, BEVDet4D (Huang and Huang 2022), and a 4D occupancy forecasting method, Cam4DOcc (Ma et al. 2024a). To ensure a fair comparison, the same GT is adopted for training. We test these methods under varied inference window, i.e., 3 and 6 frames. The 3D methods perform occupancy networks intermittently with inference delay and reused the delayed results in the inference window. BEVDet4D is adapted to process iteratively, using previous latent features to forecast future states with a 3D semantic head. We also modify the prediction head of the Cam4DOcc model to align with our settings: to predict the next 3 or 6 frames in the future. We pair CMOP with two methods of the "detection" scheme (MonoScene and SurroundOcc).

The results presented in Table 2 show that 3D occupancy prediction methods exhibit low accuracy, as they do not perform continuous updating during inference intervals. This makes them strongly affected by outdated data. In contrast, methods like BEVDet4D, Cam4DOcc, and CMOP, which incorporate historical data, achieve significantly better performance. In particular, BEVDet4D enhances performance at each timestep by using historical latent features, while Cam4DOcc predicts future occupancy only based on historical camera images without considering continuous visual changes. Conversely, compared to history-aware predicting (BEVDet4D) and implicit forecasting (Cam4DOcc), our SurroundOcc+CMOP performs the best and brings performance gains in both $\overline{\text{IoU}}$ and $\overline{\text{F1}}$ across two inference delay settings, owing to updating occupancy using both historical prediction and continuous visual changes.

Ablation study

Table 3 illustrates the impact of each module on the performance improvement over the baseline with an inference window of 6 frames. **M0** denotes the OccInfer baseline that simply performs SurroundOcc (Wei et al. 2023b) periodically with a 6-frame interval and ignores continuous updating. For frames between inferences, **M0** reuses the most recent inference result. It is clear that **M0** suffers from long inference delays and thus produces relatively low accuracy. We use it to show a lower bound of accuracy as a result of having no predictive ability. **M1** and **M2** denote predictive methods that utilize historical data and real-time visual changes, respectively. Their respective results validate the effectiveness of **M1** and **M2**. Then, Propagator and Tracker combined can enhance the accuracy of occupancy based on the adaptive fusion of Fuser, as shown in **M3**. Finally, in **M4**, we use the results from Fuser every 6 frames as the historical occupancy input in Propagator (i.e., removing OccInfer). This led to a drop in performance, indicating that periodically using OccInfer can effectively prevent error accumulation during prolonged updating.

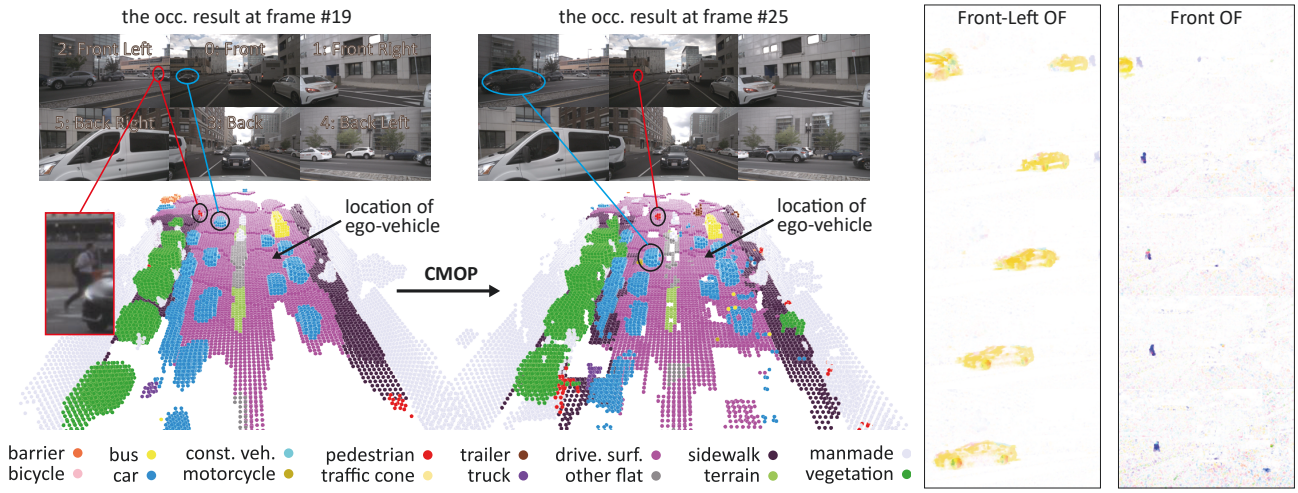
Idx.	OccInfer	P	T	F	$\overline{\text{IoU}} \uparrow$	$\overline{\text{F1}} \uparrow$
M0	✓	✗	✗	✗	21.10	34.46
M1	✓	✓	✗	✗	25.56	40.49
M2	✓	✗	✓	✗	26.38	43.61
M3	✓	✓	✓	✓	31.56	46.71
M4	✗	✓	✓	✓	29.41	45.12

Table 3: Ablation study of the key modules in our CMOP. (P=Propagator, T=Tracker, F=Fuser)

Idx.	Type of visual info.	$\overline{\text{IoU}} \uparrow$	$\overline{\text{F1}} \uparrow$	Time (ms) $\downarrow$
D1	Intermediate frame	21.49	8.75	N/A
D2	Frame difference	24.86	37.19	0.6
D3	Histogram of oriented gradients	28.51	40.59	13.9
D4	Optical flow	31.56	46.71	28.4

Table 4: Ablation study of different visual information.

Table 4 depicts the impact of different dynamic visual information inputs into Tracker. Since the nuScenes dataset labels data at 2Hz, we can utilize intermediate frames between labelled timesteps to compute various types of dynamic visual information. Here, we compare four types: original intermediate frames, frame difference (Singla 2014), histogram of oriented gradients (HOG) (Dalal and Triggs 2005), and optical flow (Farnebäck 2003). It is clear that **D1** has the lowest performance. This is mainly due to the abundance of irrelevant information in intermediate frames, making it challenging for the visual encoder to extract precise motion features, especially given the compact size requirements of the visual encoder. Frame difference provides residual motion images but still struggles with distilling irrelevant distractors from the image, resulting in a suboptimal performance as shown in **D2**. In **D3**, we concatenate sequential HOGs from static image patches to describe motion directions, offering a compact and concise motion rep-



(a) Occupancy results. (b) Optical flow.

Figure 4: An example of CMOP.

resentation. While HOG shows some improvement over **D2**, its performance remains limited due to the less detailed motion features and difficulty in correlating consecutive frames. Therefore, we employ optical flow (**D4**), which achieves the best performance in temporal accuracy with pixel-level motion directions while maintaining a satisfactory computation time of less than 30ms.

Efficiency analysis

The last column of Table 1 shows the inference time on Nvidia RTX 3090. All baseline models operate under the same settings, receiving input from six multi-cameras, each with a resolution of $1600 \times 900 \times 3$. The output occupancy dimension remains fixed at $200 \times 200 \times 16$. When $N_{\text{infer}} = 1$, CMOP takes slightly more time than the baselines, requiring only about 30 ms. Since our method works in a plug-and-play fashion, when combined with the baseline method, it increases the time overhead by no more than 5%.

Besides, to highlight our efficiency and feasibility, we deploy it alongside SurroundOcc (Wei et al. 2023b) on the Nvidia Jetson AGX Orin, which is widely used in real-world autonomous vehicles and edge services. The inference time of SurroundOcc is 0.912 seconds, whereas CMOP takes only 0.068 seconds. This means that within the inference window of SurroundOcc, our method can perform over 13 steps of continuous updating. In addition, due to our lightweight design principle, the Orin board can readily accommodate the two models running concurrently.

Case analysis

In Figure 4, we demonstrate an example of CMOP on an urban road. We conduct CMOP between the 19th frame and the 25th frame, with $N_{\text{infer}} = 6$. Figure 4a displays the occupancy results at the 19th and 25th frames, while Figure 4b shows the sequential optical flow images captured by the front-left and front cameras during this period.

According to the definition of vector field of optical flow (Beauchemin and Barron 1995), the vehicle in Figure 4b shows a quick leftward departure from the frame, while the pedestrian gradually moves to the right. By leveraging such explicit motion clues, it is clear that the jaywalker and the vehicle on the opposite lane (marked in circles) are visually aligned with the perception with accurate relative positions. We also note that additional noisy voxels appear on the right lane after updating of CMOP, thus further optimization is needed.

Conclusion

We present a new occupancy inference approach (CMOP) from a continuous updating scheme. Different from existing methods, we combine both historical occupancy results and continuous motion clues to track occupancy changes in a fast and iterative way. CMOP is a lightweight solution (CMOP) which can work like a plug-and-play module that works with other baselines. Within CMOP, Propagator and Tracker respectively predict occupancy based on two unique motion characteristics: historical motion patterns and continuous visual changes. Subsequently, Fuser adaptively aggregates results from Propagator and Tracker to generate a more robust and accurate occupancy. Experimental results validate the effectiveness of CMOP, highlighting its superior balance between performance improvement, time efficiency, and enhanced temporal accuracy.

Limitations and future work Our continuous updating approach fundamentally relies on the implicit correlation between occupied voxels and optical flow features. However, similar to the prevalent works, the instance-level motion features are not factored in, potentially leading to the "blank holes" within the object instances. At this stage, the relationship between object instances and voxels presents a challenging that we plan to tackle in our future research.

Acknowledgements

The study was funded by the National Natural Science Foundation of China under contracts No. 62422602, No. 62425101, No. 62332002, No. 62372010, No. 62027804, No. 62088102, Key Laboratory Grants 241-HF-D05-01, Shenzhen Science and Technology Program under grant number JCYJ20241202130025030, and the major key project of the Peng Cheng Laboratory (PCL2021A13). Computing support was provided by Pengcheng Cloudbrain.

References

Beauchemin, S. S.; and Barron, J. L. 1995. The computation of optical flow. *ACM computing surveys (CSUR)*, 27(3): 433–466.

Bozic, A.; Palafox, P.; Thies, J.; Dai, A.; and Nießner, M. 2021. Transformerfusion: Monocular rgb scene reconstruction using transformers. *Advances in Neural Information Processing Systems*, 34: 1403–1414.

Bradski, G. 2000. The opencv library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, 25(11): 120–123.

Bradski, G.; and Kaehler, A. 2008. *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.".

Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2020. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11621–11631.

Cao, A.-Q.; and de Charette, R. 2022. Monoscene: Monocular 3d semantic scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3991–4001.

Casas, S.; Sadat, A.; and Urtasun, R. 2021. Mp3: A unified model to map, perceive, predict and plan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14403–14412.

Corp., N. 2024. Taking Autonomous Vehicle Occupancy Prediction into the Third Dimension.

Dalal, N.; and Triggs, B. 2005. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, 886–893. Ieee.

Farnebäck, G. 2003. Two-frame motion estimation based on polynomial expansion. In *Image Analysis: 13th Scandinavian Conference, SCIA 2003 Halmstad, Sweden, June 29–July 2, 2003 Proceedings 13*, 363–370. Springer.

Gao, H.; Mao, W.; and Liu, M. 2023. VisFusion: Visibility-aware Online 3D Scene Reconstruction from Videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17317–17326.

Goutte, C.; and Gaussier, E. 2005. A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. In *European Conference on Information Retrieval (ECIR)*, 345–359.

Hu, A.; Murez, Z.; Mohan, N.; Dudas, S.; Hawke, J.; Badrinarayanan, V.; Cipolla, R.; and Kendall, A. 2021. Fiery: Future instance prediction in bird's-eye view from surround monocular cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 15273–15282.

Huang, J.; and Huang, G. 2022. Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. *arXiv preprint arXiv:2203.17054*.

Huang, Y.; Zheng, W.; Zhang, Y.; Zhou, J.; and Lu, J. 2023. Tri-perspective view for vision-based 3d semantic occupancy prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9223–9232.

Inc., M. G. 2020. An Hour with Amnon - Autonomous Vehicles Powered by Mobileye.

Inc., T. 2021. Tesla AI Day.

Ju, J.; Tseng, C. W.; Bailo, O.; Dikov, G.; and Ghafoorian, M. 2023. DG-Recon: Depth-Guided Neural 3D Scene Reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 18184–18194.

Khurana, T.; Hu, P.; Held, D.; and Ramanan, D. 2023. Point cloud forecasting as a proxy for 4d occupancy forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1116–1124.

Li, H.; Li, Z.; Akmandor, N. Ü.; Jiang, H.; Wang, Y.; and Padır, T. 2023a. Stereovoxelnet: Real-time obstacle detection based on occupancy voxels from a stereo camera using deep neural networks. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 4826–4833. IEEE.

Li, Y.; Gou, C.; and Tan, G. 2024. Taming Uncertainty in Sparse-view Generalizable NeRF via Indirect Diffusion Guidance. *arXiv preprint arXiv:2402.01217*.

Li, Y.; Wang, S.; and Tan, G. 2024. ID-NeRF: Indirect diffusion-guided neural radiance fields for generalizable view synthesis. *Expert Systems with Applications*, 126068.

Li, Y.; Yu, Z.; Choy, C.; Xiao, C.; Alvarez, J. M.; Fidler, S.; Feng, C.; and Anandkumar, A. 2023b. Voxformer: Sparse voxel transformer for camera-based 3d semantic scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9087–9098.

Li, Z.; Wang, W.; Li, H.; Xie, E.; Sima, C.; Lu, T.; Qiao, Y.; and Dai, J. 2022. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *European conference on computer vision*, 1–18. Springer.

Li, Z.; Yu, Z.; Austin, D.; Fang, M.; Lan, S.; Kautz, J.; and Alvarez, J. M. 2023c. Fb-occ: 3d occupancy prediction based on forward-backward view transformation. *arXiv preprint arXiv:2307.01492*.

Liang, T.; Xie, H.; Yu, K.; Xia, Z.; Lin, Z.; Wang, Y.; Tang, T.; Wang, B.; and Tang, Z. 2022. Bevfusion: A simple and robust lidar-camera fusion framework. *Advances in Neural Information Processing Systems*, 35: 10421–10434.

Liu, R.; Wang, W.; and Yang, Y. 2024. Volumetric Environment Representation for Vision-Language Navigation. *arXiv preprint arXiv:2403.14158*.

Lu, F.; Chen, G.; Li, Z.; Zhang, L.; Liu, Y.; Qu, S.; and Knoll, A. 2021. Monet: Motion-based point cloud prediction network. *IEEE Transactions on Intelligent Transportation Systems*, 23(8): 13794–13804.

Ma, J.; Chen, X.; Huang, J.; Xu, J.; Luo, Z.; Xu, J.; Gu, W.; Ai, R.; and Wang, H. 2024a. Cam4DOcc: Benchmark for Camera-Only 4D Occupancy Forecasting in Autonomous Driving Applications. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Ma, Q.; Tan, X.; Qu, Y.; Ma, L.; Zhang, Z.; and Xie, Y. 2024b. COTR: Compact Occupancy TRansformer for Vision-based 3D Occupancy Prediction. *CVPR*.

Mahjourian, R.; Kim, J.; Chai, Y.; Tan, M.; Sapp, B.; and Anguelov, D. 2022. Occupancy flow fields for motion forecasting in autonomous driving. *IEEE Robotics and Automation Letters*, 7(2): 5639–5646.

Mersch, B.; Chen, X.; Behley, J.; and Stachniss, C. 2022. Self-supervised point cloud prediction using 3d spatio-temporal convolutional networks. In *Conference on Robot Learning*, 1444–1454. PMLR.

Ming, Z.; Berrio, J. S.; Shan, M.; and Worrall, S. 2024. InverseMatrixVT3D: An Efficient Projection Matrix-Based Approach for 3D Occupancy Prediction. *arXiv preprint arXiv:2401.12422*.

Murez, Z.; Van As, T.; Bartolozzi, J.; Sinha, A.; Badrinarayanan, V.; and Rabinovich, A. 2020. Atlas: End-to-end 3d scene reconstruction from posed images. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, 414–431. Springer.

Pan, M.; Liu, J.; Zhang, R.; Huang, P.; Li, X.; Liu, L.; and Zhang, S. 2023. Renderocc: Vision-centric 3d occupancy prediction with 2d rendering supervision. *arXiv preprint arXiv:2309.09502*.

Singla, N. 2014. Motion detection based on frame difference method. *International Journal of Information & Computation Technology*, 4(15): 1559–1565.

Sun, J.; Xie, Y.; Chen, L.; Zhou, X.; and Bao, H. 2021. NeuralRecon: Real-time coherent 3D reconstruction from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15598–15607.

Tang, P.; Wang, Z.; Wang, G.; Zheng, J.; Ren, X.; Feng, B.; and Ma, C. 2024. Sparseocc: Rethinking sparse latent representation for vision-based semantic occupancy prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15035–15044.

Van Houdt, G.; Mosquera, C.; and Nápoles, G. 2020. A review on the long short-term memory model. *Artificial Intelligence Review*, 53(8): 5929–5955.

Wang, S.; Xu, H.; Li, Y.; Chen, J.; and Tan, G. 2024. IE-NeRF: Exploring transient mask inpainting to enhance neural radiance fields in the wild. *Neurocomputing*, 129112.

Wei, Y.; Zhao, L.; Zheng, W.; Zhu, Z.; Rao, Y.; Huang, G.; Lu, J.; and Zhou, J. 2023a. Surrounddepth: Entangling surrounding views for self-supervised multi-camera depth estimation. In *Conference on Robot Learning*, 539–549. PMLR.

Wei, Y.; Zhao, L.; Zheng, W.; Zhu, Z.; Zhou, J.; and Lu, J. 2023b. Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 21729–21740.

Xu, G.; Yin, W.; Chen, H.; Shen, C.; Cheng, K.; and Zhao, F. 2023. Frozenrecon: Pose-free 3d scene reconstruction with frozen depth models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 9276–9286. IEEE.

Yan, X.; Gao, J.; Li, J.; Zhang, R.; Li, Z.; Huang, R.; and Cui, S. 2021. Sparse single sweep lidar point cloud segmentation via learning contextual shape priors from scene completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 3101–3109.

Yang, L.; Yu, K.; Tang, T.; Li, J.; Yuan, K.; Wang, L.; Zhang, X.; and Chen, P. 2023. BEVHeight: A Robust Framework for Vision-based Roadside 3D Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21611–21620.

Yu, C.; Wang, J.; Gao, C.; Yu, G.; Shen, C.; and Sang, N. 2020. Context prior for scene segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12416–12425.

Yu, Z.; Shu, C.; Deng, J.; Lu, K.; Liu, Z.; Yu, J.; Yang, D.; Li, H.; and Chen, Y. 2023. Flashocc: Fast and memory-efficient occupancy prediction via channel-to-height plugin. *arXiv preprint arXiv:2311.12058*.